

(21) Application No **0006487.3**

(22) Date of Filing **18.03.2000**

(30) Priority Data

(31) **9925003**

(32) **22.10.1999**

(33) **GB**

(31) **9925004**

(32) **22.10.1999**

(71) Applicant(s)

Roke Manor Research Limited
(Incorporated in the United Kingdom)
Roke Manor Research, Old Salisbury Lane, ROMSEY,
Hampshire, SO51 0ZN, United Kingdom

(72) Inventor(s)

Alan Mark Burnett
David Bryan Keogh

(74) Agent and/or Address for Service

Siemens Shared Services Limited
Intellectual Property Department, Siemens House,
Oldbury, BRACKNELL, Berkshire, RG12 8FZ,
United Kingdom

(51) INT CL⁷

G06F 13/12 , G08C 17/02 19/28 23/04

(52) UK CL (Edition S)

G4A AFGDC

G4H HRBE HRCE H1A H13D H14A H14D H14G H60

U1S S1820 S1912 S1931 S1966 S1978 S2105 S2106

S2107 S2124 S2125 S2127 S2146 S2179 S2188 S2206

S2210 S2215

(56) Documents Cited

GB 2309567 A

GB 2304952 A

WO 96/35286 A1

(58) Field of Search

UK CL (Edition S) **G4A AFGDC AFGDX , G4H HNEC**

HNEM HNP HRCE

INT CL⁷ **G06F , G08C**

ONLINE:WPI,EPODOC,JAPIO

(54) Abstract Title

A fully integrated Web activated control and monitoring device

(57) A web-enabled microcontroller device is provided with both web server functions (310) and generic control and monitoring functions (312). The web-enabled microcontroller device may be embedded in domestic, commercial and industrial hardware. Integrated software for remotely controlling hardware by means of the microcontroller device combines control application code (312) and HTTP server code (310). One implementation of the microcontroller device has a microprocessor coupled to a physical communications unit, a ROM and a RAM. The protocol stack (314) associated with the HTTP server may be permanently coded into the ROM or loaded into the RAM as required. In another implementation, the physical communications unit includes a digital signal processor and a wireless access unit, thereby providing a web-enabled digital wireless access device. In the wireless implementation, the processing of the wireless access physical layer (302) is performed by the digital signal processor and the higher layer processing (310, 312, 314) is performed on the microprocessor.

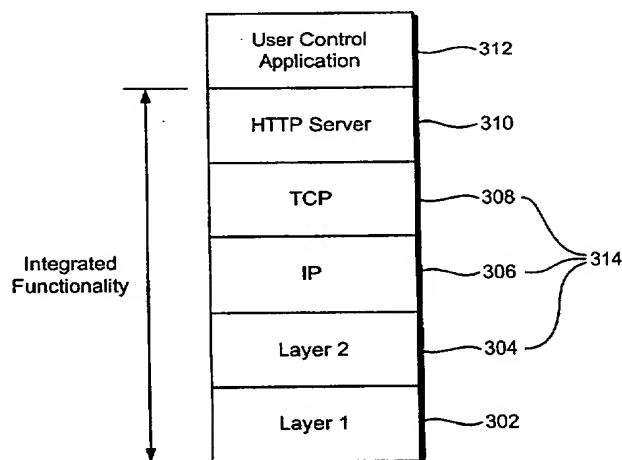
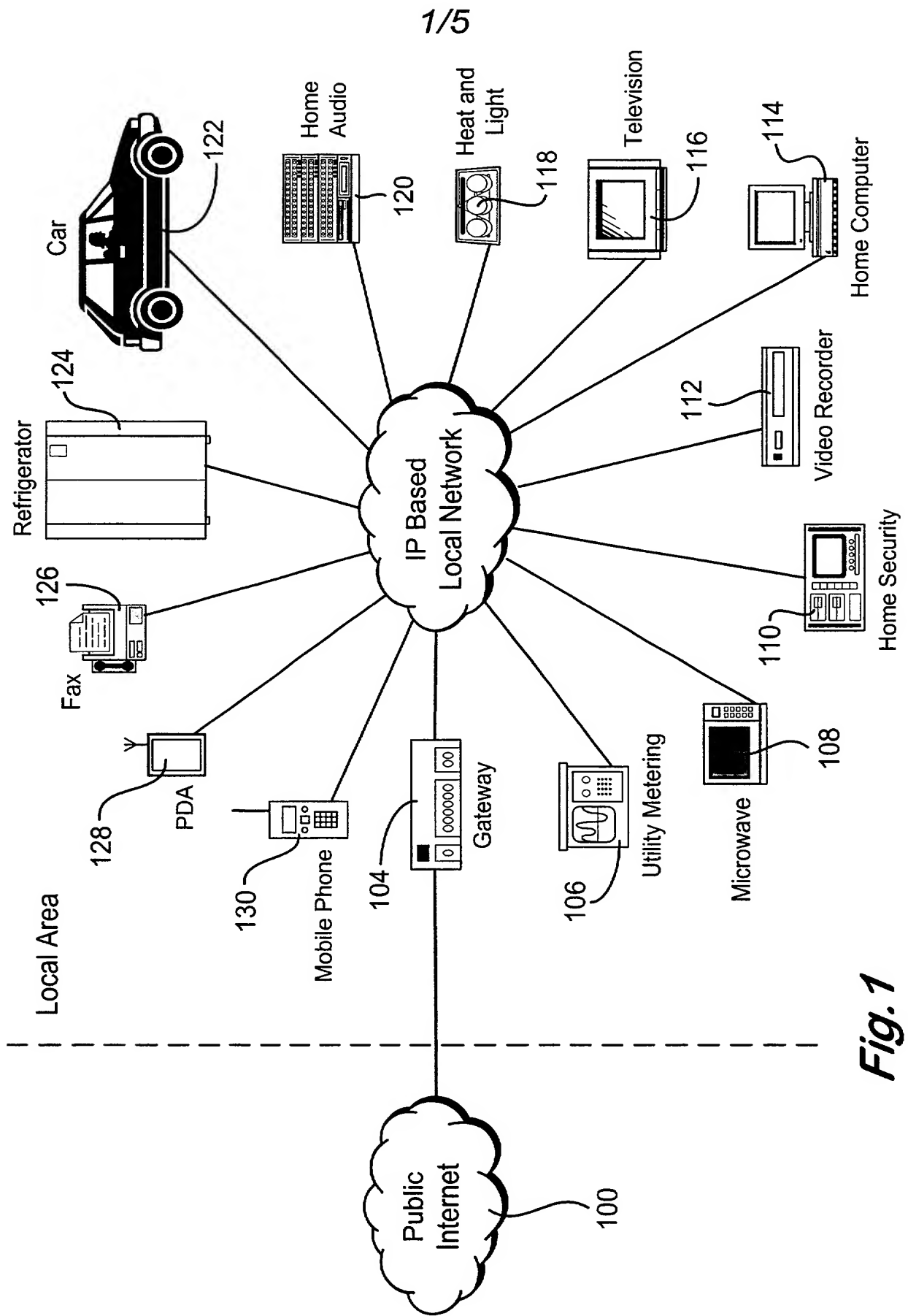


Fig.3



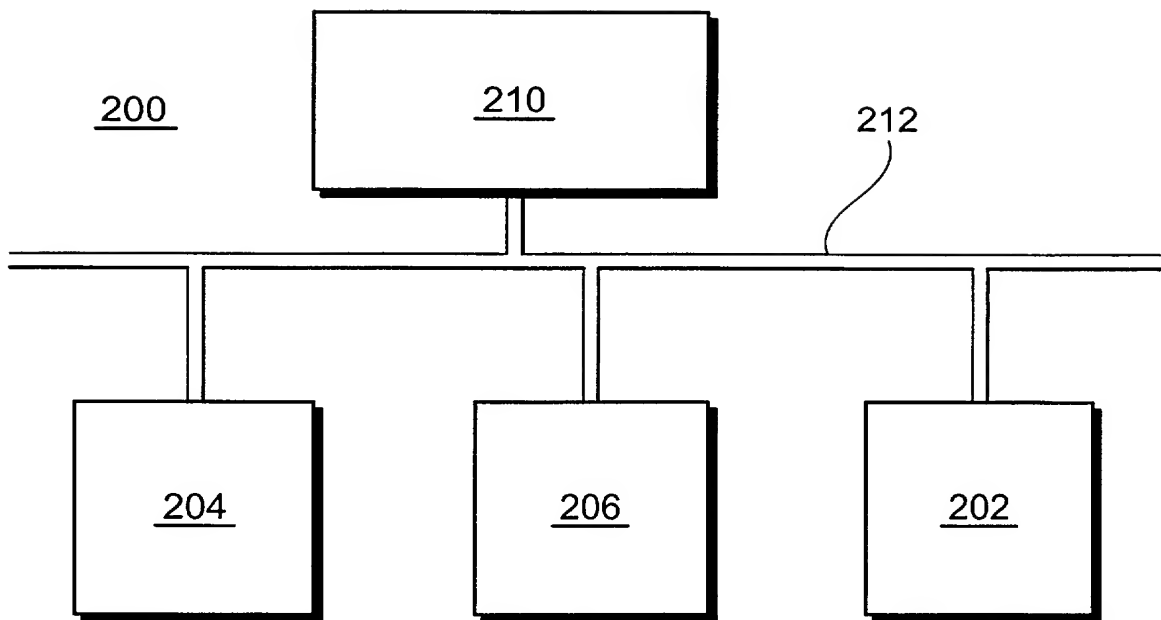


Fig. 2A

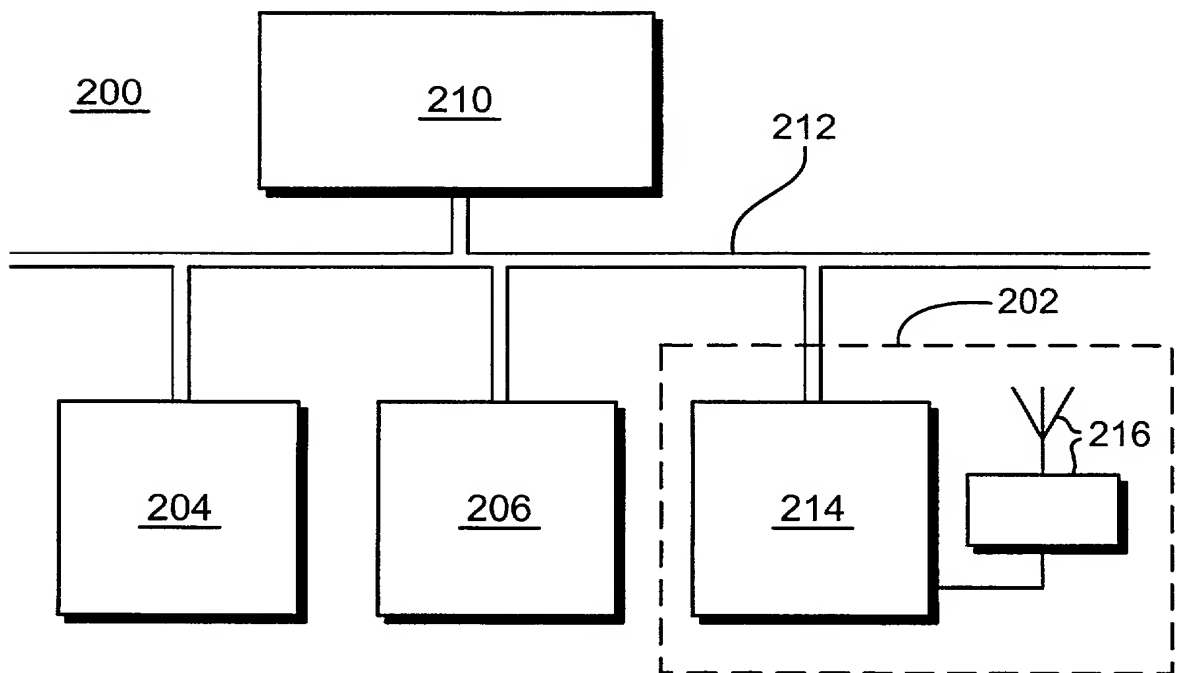
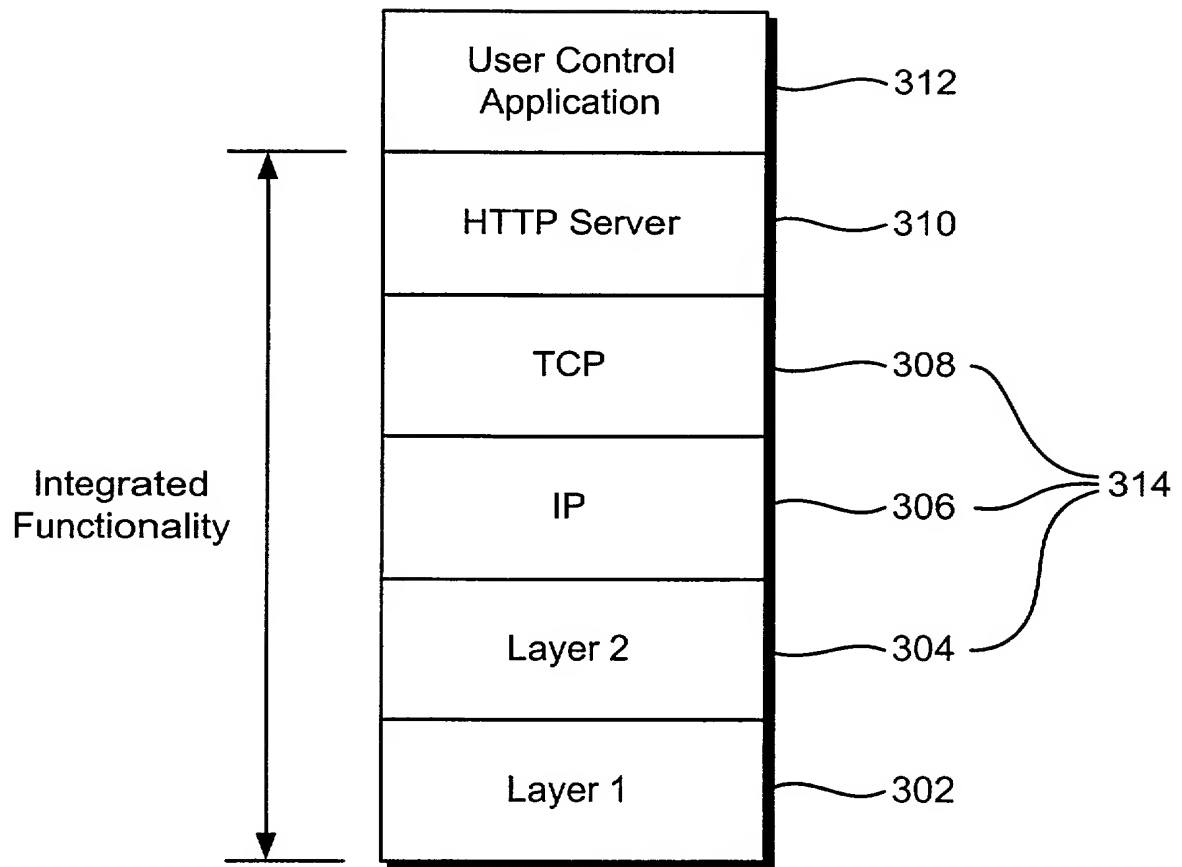


Fig. 2B

***Fig. 3***

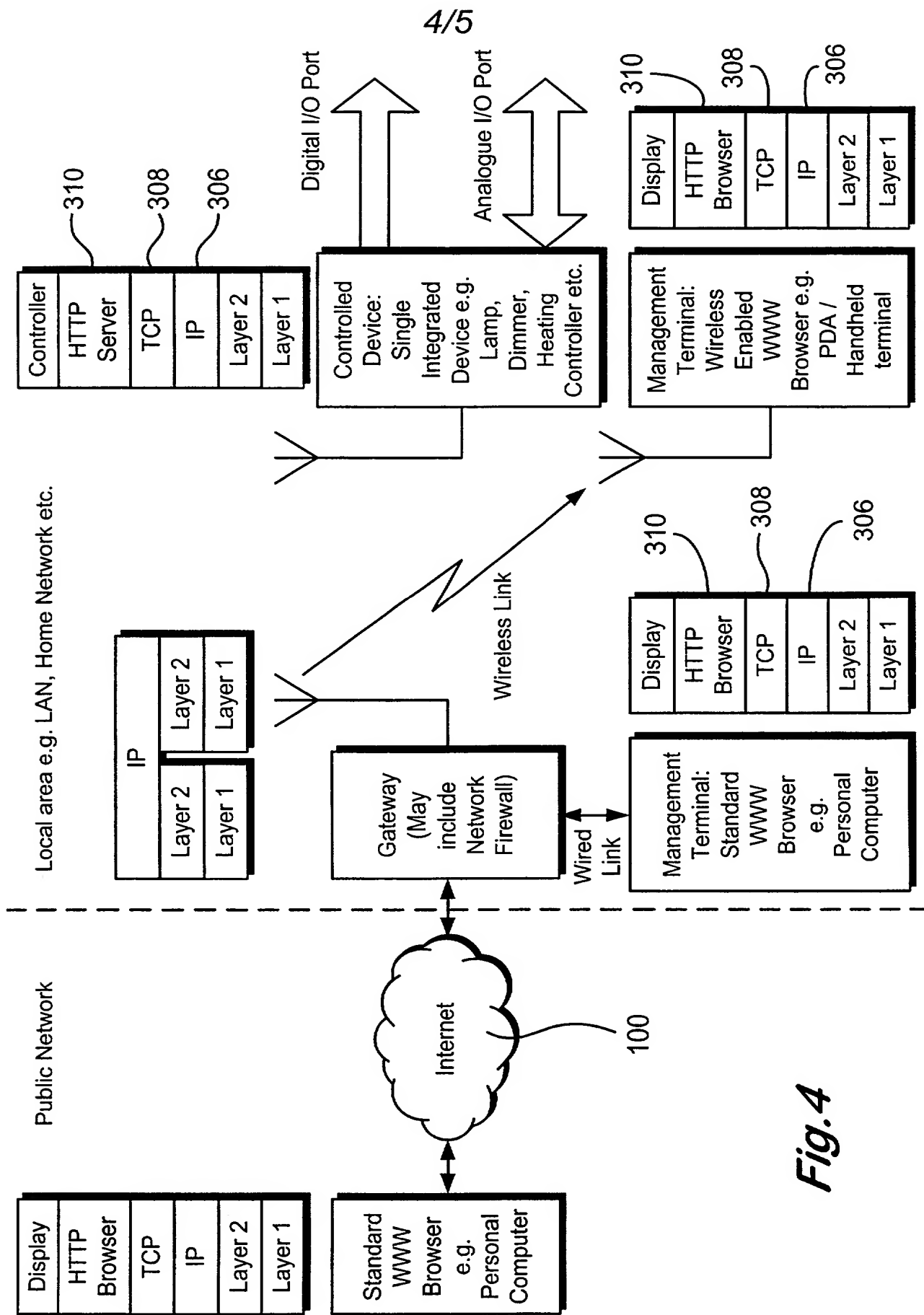
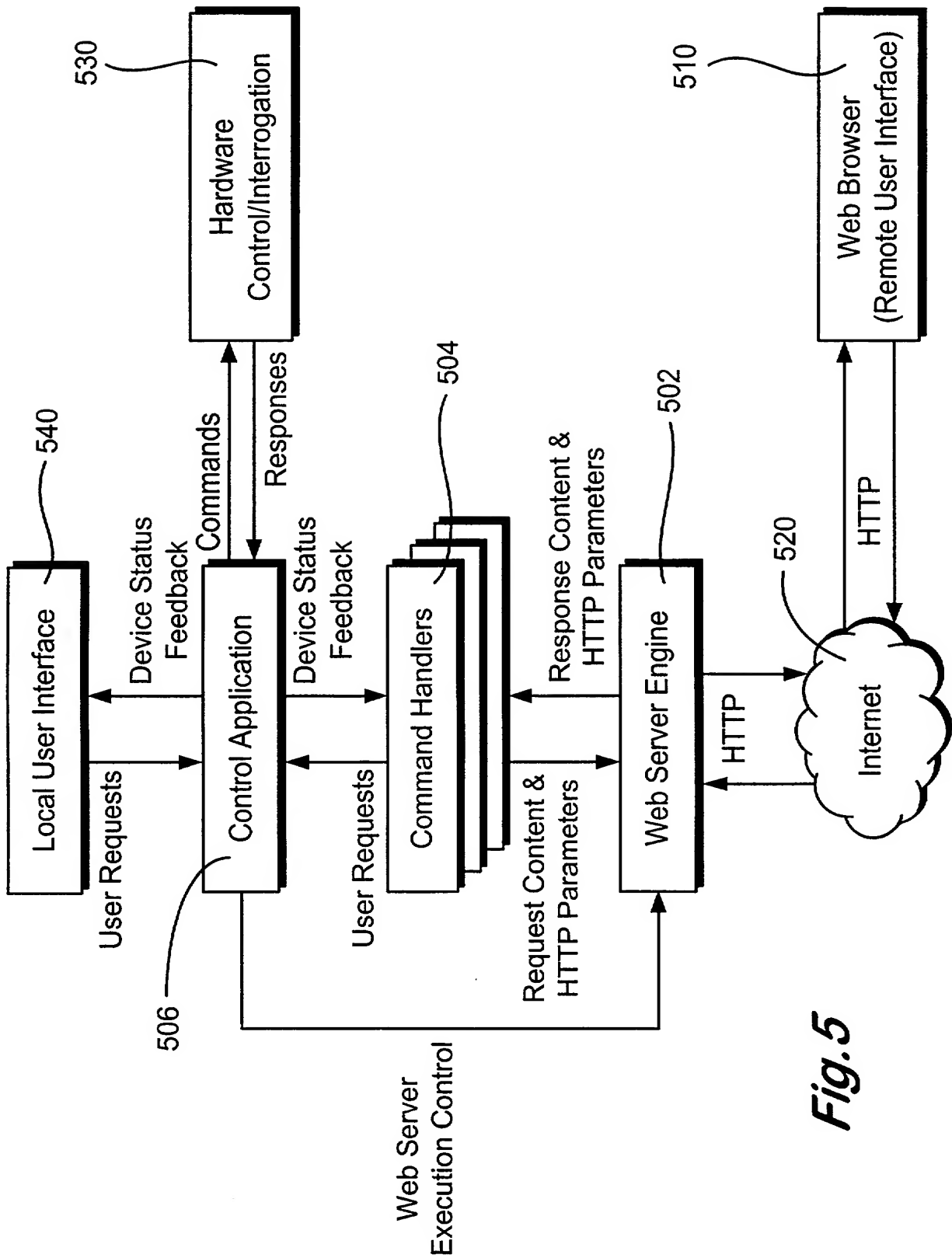


Fig. 4

**Fig.5**

A FULLY INTEGRATED WEB ACTIVATED CONTROL AND MONITORING DEVICE

This invention relates to a fully integrated, Web activated control and
5 monitoring device. The control device of the invention can be used for
remotely controlling and monitoring functions in a physical device.

The Internet and World Wide Web are now ubiquitous technologies for
information transfer. One prominent data networking standard for
information transfer is TCP/IP (Transmission Control Protocol / Internet
10 Protocol). The majority of home computers are now supplied 'Internet
Ready' and 'Web-enabled', ensuring that many consumers are not only aware
of the technology, but already have it in their homes.

Web servers deliver content according to a given protocol for
presentation on a remote client device, using a Web browser. In the original
15 World Wide Web (WWW) concept, the given protocol was a Hypertext
Transfer Protocol (HTTP). An HTTP server delivers Hypertext Markup
Language (HTML) pages to one or more users, for display by an HTTP
browser. The HTML pages are static files containing text and image
information. This concept has evolved to include dynamic HTML, whereby
20 interactive pages (having radio buttons and drop down menus for example)
provide an interface to other applications such as search engines, databases or
online dictionaries. Web servers have evolved so that in addition to
supporting HTTP, the Web servers can support other information transfer
protocols including a wireless application protocol (WAP), the WAP being
25 particularly appropriate for delivering content in wireless implementations.

By embedding a Web server in a target physical device, the WWW
concept can be further extended to include the control and monitoring of the
physical device. Examples of suitable target physical devices include video

recorders, central heating systems and home security systems. The device being controlled can present a protocol-compliant interface to any network using IP (for example the public Internet, an Intranet or a home network) and can thus interface with any Web browser. The structure and content of the protocol-compliant interface presented to the Web browser is stored entirely on the controlled device.

Consider the case of an HTTP server which can be linked to a control application. The control application in turn interfaces with hardware components of the physical device. Examples of hardware components which can be controlled in this manner include: Digital to Analogue Converters (DAC); Analogue to Digital Converter (ADC); parallel Input/Output (I/O) or serial I/O devices; display devices; and device monitoring logic. Rather than simply serving fixed HTML pages, the HTTP server may deliver control and monitoring information using dynamic Web page graphical constructs, for example radio buttons or pull-down menus. Consequently, the control application provides the HTTP server with device status information in HTML format and HTML formatted menus for the setting of device control parameters. The HTTP content presented to a user may be a simulation (or exact copy) of the interface presented by the physical device being controlled.

In general, Web servers have been designed to operate on high performance, general-purpose computing platforms, and are optimised for serving content pages at high speed to a large number of users. It is known to provide HTTP servers for control applications by embedding a general-purpose computer in the physical device to be controlled or by controlling a number of local devices from a single computer. However, such an approach is clearly unsuitable for consumer applications due to the high cost of implementation.

Alternatively physical devices can be controlled by embedded microprocessors (or microcontrollers) rather than general-purpose computers; Web servers suitable for the embedded microprocessors have been produced. Typically, these Web servers are not fundamentally different in design to those used in larger scale general-purpose computing environments. Generally, the Web server software for the embedded microprocessors is smaller and supports fewer features. However, Web servers for general purpose computers and servers for the embedded microprocessors do still have the following features in common.

Firstly, both types of Web server are implemented as a separate piece of code that executes in a separate thread or process alongside the control application code.

Secondly, an external scheduling system (usually part of an operating system) determines when both the Web server software and the control application software get execution time.

Thirdly, as the control application code and the Web server code are both designed to execute separately, code needed to perform standard tasks will often be duplicated in both, adding to the overall size of the software.

Lastly, both types of Web server code incorporate code which handles differences between protocols. In the case of the HTTP, there is no single standard definition, rather there are multiple standard versions, each of which has many optional features. In addition, there are a number of non-HTTP features that are widely considered to be server options. Since a Web server is typically a separate piece of code, designed to be applied across a wide range of applications, the Web server usually contains code to support functionality that is not used by a given specific application. The implementation of a Web server in an embedded environment as described above is not optimal in a typical consumer appliance implementation using a

low cost microcontroller. The memory and processing power available in typical consumer appliances are very restricted.

Ideally, the Web server software for implementation in embedded environments should include no redundant code and the control application software should be able to dictate when any support software, such as the Web server software, receives execution time. When this control application software is implemented in embedded microcontroller devices, the embedded microcontroller devices may be used to replace existing, known microcontroller devices. In addition, the cost and inconvenience of wiring is a significant barrier to home networking. The implementation of a wireless environment is thus desirable and so there is a need for Web server technology in the context of digital radio devices in order to produce a class of physical device particularly appropriate for home networking implementations.

According to the present invention, there is provided a microcontroller device for controlling at least one physical device, the microcontroller device including a microprocessor coupled to a physical communication unit and a memory unit, wherein the memory unit stores an integrated piece of software arranged to perform a server function and a control application function and to support protocol stacks.

Preferably, the physical communication unit includes a UART chip. Equally preferably, the physical communication unit includes a digital signal processor and a wireless access unit. More preferably the digital signal processor is a baseband processor.

Preferably, the microcontroller device is embedded in a single integrated circuit.

Preferably the memory unit includes at least one non-volatile memory unit, the memory unit advantageously, permanently storing the integrated

piece of software. More preferably, the non-volatile memory is read only memory (ROM) unit.

Alternatively, the memory unit includes at least one volatile memory unit. The integrated software is preferably stored as an image on the at least one volatile memory unit. The at least one volatile memory unit may be a random access memory (RAM) unit.

Preferably, the server functions provided by the integrated piece of software comply with a communications protocol. More preferably, the communications protocol is a Hypertext Transfer Protocol and protocol stacks supported by the integrated piece of software are compatible with the Hypertext Transfer Protocol. Equally preferably, the communications protocol is a Wireless Application Protocol and protocol stacks supported by the integrated piece of software are compatible with the Wireless Application Protocol.

Preferably, there is provided a physical device controlled by the microcontroller device. More preferably, there is provided a communications network comprising at least one physical device controlled by the microcontroller device.

According to the present invention, there is also provided a method of remotely controlling a physical device having at least one function, the method including the steps of: providing the physical device with a server engine for receiving incoming communications from a remote user; providing the physical device with a command handler arranged to produce an interpreted remote user communication by interpreting the incoming communication; providing the physical device with at least one control application arranged to receive the interpreted remote user communication and to interface with the physical device for controlling the at least one

function; and controlling the at least one function in response to the at least one control application.

Preferably, a memory unit having an integrated piece of software recorded thereon is provided, wherein the integrated piece of software performs the method above.

At least one embodiment of the present invention will now be described, by way of example only, with reference to the following drawings, in which:

Figure 1 is a schematic diagram of an IP-based home network;

Figure 2A is a schematic diagram of the layout of a first web-enabled microcontroller device;

Figure 2B is a schematic diagram of the layout of a second web-enabled microcontroller device for use in a wireless network environment;

Figure 3 is a first reference model for the first web-enabled generic microcontroller device of Figure 2A;

Figure 4 is a schematic diagram of a network showing wireless and wired inter-working between devices using different physical layer media; and

Figure 5 is a schematic diagram of the integration of an HTTP server with a control application.

Throughout the following description, identical reference numerals are used to identify like parts.

Referring to Figure 1, an IP based home network 102 is coupled to a plurality of physical devices, the IP based home network 102 being in communication with the public Internet 100 via an IP gateway 104. The plurality of physical devices can include: computing devices, such as a home computer 114 or a Personal Digital Assistant (PDA) 128; home entertainment devices, such as a television 116, a video recorder 112, home audio

equipment 120; domestic appliances, such as a refrigerator 124, a fax machine 126, a mobile telephone 130, a microwave oven 108 or heating and lighting facilities 118; security equipment 110; a utility meter 106, such as a gas meter or an electricity meter; and automotive monitoring equipment such as car engine management equipment 122.

Remote control of the plurality of physical devices requires that each physical device has a microcontroller arrangement for control and monitoring of the physical device, and embedded HTTP server software. The HTTP server software permits communication with a management terminal, for example the home computer 114 or the wireless web enabled PDA 128. The plurality of physical devices are managed using an HTTP browser implemented upon the management terminal in order to present information generated by the HTTP server software in the form of content pages. In order to manage the plurality of physical devices, the management terminal requires respective IP addresses associated with each of the plurality of physical devices. The respective IP addresses can be obtained by an automatic discovery process, for example the Bluetooth 'Service Discovery Protocol'.

Although reference has been made to "Bluetooth" it should be appreciated that other wireless access techniques, for example, Infrared Data Association (IrDA) or Shared Wireless Access Protocol (SWAP), can be used to reduce the cost and complexity of network wiring.

In a first embodiment of the invention (Figure 2A), a web-enabled microcontroller arrangement 200 includes a physical communications unit 202 coupled to a microprocessor 210 via a first data bus 212, a non-volatile memory unit 204 (e.g. ROM, EEPROM, EPROM or flash memory) and a volatile memory unit 206 (e.g. RAM or random access flash) also being coupled to the first data bus 212. The microcontroller arrangement 200 is suitable for connection to a network for communications with remote devices

coupled to the network by means of the physical communications unit 202, for example, a Universal Asynchronous Receiver/Transmitter chip (UART).

Referring to Figure 3, the web-enabled microcontroller arrangement 200 implements an integrated protocol stack 314 providing a data link layer (Layer 2) 304, an IP layer 306 and a TCP layer 308, the IP layer 306 and the TCP layer 308 providing transmission and networking layers. The integrated protocol stack 314 is overlaid upon a physical layer 302 (Layer 1). An HTTP layer 310 is overlaid upon the TCP layer 308, the HTTP layer 310 being implemented by HTTP server software. The HTTP server software, the integrated protocol stack 314 and user control application software 312 all execute on the embedded microprocessor 210. The physical layer 302, the data link layer 304, the IP layer 306, the TCP layer 308 and the HTTP layer 310 are either coded permanently into the non-volatile memory unit 204 (for example, mask programmed onto ROM) or provided as an image which may be loaded by a software developer into the volatile memory unit 206. In the latter case, the software developer is free to generate control application software specific to the hardware of the controlled device and linking directly to the HTTP server software.

The code build process incorporates the building of both control application code and HTTP server code to give a single piece of web-enabled software. This approach gives a number of advantages in terms of overall size and complexity of the web-enabled software, and in terms of utilisation of available processing power.

Firstly, the HTTP server software presents a generic user interface with content dependent only on the controlled device itself. The HTTP server software uses a networking standard (TCP/IP) which allows a device to be monitored or controlled from anywhere in the world with no translation whatsoever of the content presented by the HTTP server. The device can be

controlled from a known platform; for example, a workstation, the home personal computer 114 or the PDA 128.

Secondly, the HTTP server code and the control application code are combined to form a single piece of software in which the HTTP server code only executes when called by the control application code. By structuring the HTTP server code so that it periodically returns thread mastery to the control application code throughout the processing of an incoming HTTP communication, and by providing parameters to configure this processing both dynamically and as part of the code build, the application is given a high degree of control over the available processing power. When active, the HTTP server can be seen as making use of spare capacity not used for the control application.

Thirdly, since the HTTP server code is part of an implementation specific code build process, the web-enabled control application software includes code to support any and all desirable HTTP server features. By the use of standard code building techniques and suitable structuring of the HTTP server code, any HTTP server code supporting functionality that is optional, yet not relevant for the specific implementation for the physical device, can simply be excluded from the final software as part of the code build process for that implementation.

Finally, as the control application code and HTTP server code form part of the same piece of web-enabled software, they can both make use of the same sections of code to perform standard tasks rather than duplicate these sections in each case.

In a second embodiment of the present invention (Figure 2B), the microcontroller arrangement 200 differs from that of the first embodiment (Figure 2A) by the physical communication unit 202 having an embedded

DSP 214 and a radio frequency (RF) transceiver device 216, for transmitting and receiving signals.

In the context of the reference model of Figure 3, the physical layer 302 is supported by the transceiver unit 216 and the embedded DSP 214. Signals received by the transceiver unit 216 are processed by the embedded DSP 214, the embedded DSP 214 can also support the (baseband) data link layer 304 (layer 2). Although specific apparatus have been described for the full or partial support of layers 1 and 2 of the reference model, it should be appreciated that other hardware known in the art can be used to support the layers 1 and 2.

Some functionality of the data link layer 304, the IP layer 306, the TCP layer 308 and the HTTP layer 210 is provided by the embedded microprocessor 210. Typical wireless access technologies that use this device architecture include: Digital Enhanced Cordless Telecommunications (DECT), Global System for Mobile communications (GSM), Bluetooth, Universal Mobile Telecommunications System (UMTS), IrDA or SWAP.

It should be understood that the embedded microprocessor 210 can be instructed to perform the operations of the DSP 214 in addition to Layer 2 and control processing, thus the DSP 214 is not absolutely essential. Alternatively, logic circuitry can be provided to perform some or all tasks of the DSP 214 and the embedded microprocessor 210.

Referring to Figure 4, the wired and wireless inter-working between devices of Figure 1 can be seen in more detail. Layers 1 and 2 of the wired part of the network differ from layers 1 and 2 of the wireless part of the network. However, the IP layer 306, the TCP layer 308 and the HTTP server layer 310 are identical, making inter-working straightforward.

In addition to the above described applications, physical devices comprising the microcontroller arrangement 200 can be used for diagnostic

and configuration functions by a remote service centre. For example, a user can have difficulty configuring one of the physical devices or the physical device may appear to be faulty. The user can connect the physical device to the remote service centre via the public Internet 100, where skilled service personnel can run diagnostics remotely or assist the user in the configuration of the physical device. Additionally, the home entertainment equipment incorporating the embedded Web server can include specialised configuration and diagnostic functions not accessible to the user. The configuration and diagnostic functions, inaccessible to the user, can be accessed by a remote customer service centre via the public Internet 100.

The home security systems 110 can also have additional remote management features. A security company can use embedded HTTP server technology to provide additional management services to a client. The HTTP server software can be used to control and monitor audio and video surveillance equipment, and also to control building infrastructure functions, for example, heat and power systems.

In another application, the utility meter 106 can be read and controlled remotely by both a customer and a utility provider via the wired public Internet 100 or a wireless network in order to save on collection/reading costs and to provide a common user interface.

The HTTP server software embedded in the car engine management system 122 can provide a valuable diagnostic tool. The engine management system 122 collects performance and service interval information, which can be interrogated locally by a garage, or remotely over the public Internet 100 by a manufacturer's service centre or a breakdown service. Information gathered can be used for a number of purposes including: by the garage to identify items requiring adjustment/replacement during service; by the garage to identify a fault after breakdown; by the breakdown service to identify a

fault before dispatching roadside assistance; and by the manufacturer to gather performance information over the life of a vehicle.

In order to exclude unauthorised users from the home IP network 102 connected to the public Internet 100 where authorised users are relatively unskilled, a firewalling technique is employed. In general, the firewalling technique restricts access to local networks from the unauthorised users. The firewalling technique is implemented at the gateway 104 between the public Internet 100 and the home IP network 102. Access to the home IP network 102 is restricted by applying security measures known in the art including filtering of incoming packets on combinations of identification numbers including source IP addresses, destination IP addresses, TCP/IP port numbers and UDP (User Datagram Protocol) port numbers.

In a wired network the firewalling function is performed by an IP access router known in the art acting as a Local Area Network (LAN) switch/router for the controlled devices connected to it. However, in a wireless network, it is possible that the gateway is only an access point, and performs no switching or routing functions for the devices in communication with the wireless network. The firewalling technique is still required for the wireless network, but no physical media switching is needed. Consequently, the gateway 104 comprises a firewalling device implementing the firewalling technique and the protocol relay function in order to provide an access point to the public Internet 100 to a home IP network 102. In this case, the firewalling device is just another web-enabled physical device and can be managed, configured and diagnosed remotely, with expert help as required.

Referring to Figure 5, the control application 506 is capable of relaying commands to a hardware control module 530 and receives responses from the hardware control module 530. The HTTP server software includes an HTTP server engine 502 and a plurality of command handlers 504. Each command

handler 504 is a piece of application specific code that the HTTP server engine 502 uses to handle a respective application specific aspect of an HTTP communication for the control application 506.

5 The control application 506 registers each command handler 504 with the HTTP server engine 502. The HTTP server engine 502 then calls one of the plurality of command handler 504 appropriate for handling a respective application specific aspect of an HTTP communication at various stages of the HTTP communication. The registration process associates a textual name with each command handler 504, thus allowing the appropriate command
10 handler 504 to be specified within the HTTP communication (not shown). The HTTP server engine 502 looks for textual names of command handlers 504 within HTTP communications and maps each textual name found, using the list of registered command handlers, to an appropriate corresponding command handler 504.

15 Furthermore, the command handlers 504 are specific to the hardware ultimately being controlled through the hardware control module 530. Any parameters and context specific content passed by the HTTP server engine 502 to the control application 506 via a given command handler 504 are interpreted by the given command handler 504, the command handler 504
20 acting as an interface between the control application 506 and the Web server engine 502. In response to the parameters and context specific content delivered via the command handler 504, the control application 506 performs whatever action is appropriate given the nature of the HTTP communication (not shown). The command handlers 504 also interpret the results of actions
25 by the control application 506 and act as the interface back to the HTTP server engine 502, for example the command handlers 504 identify errors and/or produce parameters and context specific content for the HTTP server engine 502 to format into an outgoing HTTP communication (not shown).

The HTTP server engine 502 handles all aspects of the HTTP connection, from receiving and sending of HTTP communications, to parsing and formatting the parameters and context specific content.

The HTTP server engine 502 also handles sequencing, by calling an appropriate command handler 504 at various points throughout the process of handling an HTTP communication. The sequencing is, however, still under the overall control of the control application 506, due to other features of the HTTP server software. The control application 506 can, either as part of the build process or dynamically for each HTTP communication, specify parameters, for instance, how much data may be sent or received over the TCP/IP connection before returning execution to the control application 506. Additionally, in the sequence of events that are required to handle an HTTP communication, there are fixed points where the control application 506 can define whether the HTTP server engine 502 continues automatically with the next event of the sequence or returns execution to the control application 506 after each event. The HTTP server engine 502 therefore is only allowed execution time when called by the control application 506. The amount of execution time allocated at each call to the HTTP server engine 502 can thus be configured, as can the number of simultaneous HTTP communications that can be handled.

The user can still interact locally with the hardware of the physical device through a local user interface 540. Local user communications (not shown) are handled directly by the control application 506, the control application 506 acting as an interface to the hardware control module 530. Feedback from the hardware control module 530 returns to the user interface 540 via the control application 506.

It will be understood that, although the foregoing description is concerned with HTTP server code, other protocols can be adopted in place of

hypertext transfer protocol. Most notably the Wireless Application Protocol (WAP) is suitable for the implementation of a Web server in a wireless environment. Suitable alternative protocols within which the invention can be applied include file transfer protocol (FTP), Session Initiation Protocol (SIP), Service Location Protocol (SLP), telnet and secure HTTP (SHTTP). Suitable protocol stacks for the physical layer 302 and the data link layer (Layer 2) 304 include: X.25, AppleTalk, Ethernet and asynchronous transfer mode (ATM).

Claims:

1. A microcontroller device for controlling at least one physical device, the microcontroller device including a microprocessor coupled to a physical communication unit and a memory unit, wherein the memory unit stores an integrated piece of software arranged to perform a server function and a control application function and to support protocol stacks.
2. A microcontroller device according to Claim 1, wherein the physical communication unit includes a UART chip.
3. A microcontroller device according to Claim 1, wherein the physical communication unit includes a digital signal processor and a wireless access unit.
4. A microcontroller device according to Claim 3, wherein the digital signal processor is a baseband processor.
5. A microcontroller device according to Claims 1, 2, 3 or 4, wherein the microcontroller device is embedded in a single integrated circuit.
6. A microcontroller device according to any one of Claims 1 to 5, wherein the memory unit includes at least one non-volatile memory unit.
7. A microcontroller device according to Claim 6, wherein the at least one non-volatile memory unit permanently stores the integrated piece of software.

8. A microcontroller device according to Claims 6 or 7, wherein the at least one non-volatile memory unit is a read only memory unit.

9. A microcontroller device according to any one of Claims 1 to 6,
5 wherein the memory unit includes at least one volatile memory unit.

10. A microcontroller device according to Claim 9, wherein the integrated piece of software is stored as an image on the at least one volatile memory unit.

10 11. A microcontroller device according to Claims 9 or 10, wherein the at least one volatile memory unit is a random access memory unit.

12. A microcontroller device according to any one of the preceding
15 claims, wherein the server functions provided by the integrated piece of software comply with a communications protocol.

13. A microcontroller device according to Claim 12, wherein the communications protocol is a Hypertext Transfer Protocol and protocol stacks supported by the integrated piece of software are compatible with the
20 Hypertext Transfer Protocol.

14. A microcontroller device according to Claim 12, wherein the communications protocol is a Wireless Application Protocol and protocol stacks supported by the integrated piece of software are compatible with the
25 Wireless Application Protocol.

15. A physical device controlled by a microcontroller device as claimed in any one of the preceding claims.

5 16. A communications network comprising at least one physical device according to Claim 15.

17. A method of remotely controlling a physical device having at least one function, the method including the steps of:

10 providing the physical device with a server engine for receiving incoming communications from a remote user;

providing the physical device with a command handler arranged to produce an interpreted remote user communication by interpreting the incoming communication;

15 providing the physical device with at least one control application arranged to receive the interpreted remote user communication and to interface with the physical device for controlling the at least one function; and controlling the at least one function in response to the at least one control application.

20 18. A memory unit having an integrated piece of software recorded thereon, wherein the integrated piece of software performs the method as claimed in Claim 17.

25 19. A microcontroller device substantially as hereinbefore described with reference to the accompanying drawings.

20. A network of microcontroller devices substantially as hereinbefore described with reference to the accompanying drawings.

21. A physical device substantially as hereinbefore described with reference to the accompanying drawings.



Application No: GB 0006487.3
Claims searched: 1-16,19-21

Examiner: Mike Davis
Date of search: 15 November 2000

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AFGDC, AFGDX), G4H (HRCE)

Int Cl (Ed.7): G06F

Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	WO 96/35286 A1 (NOKIA...) whole document, especially Fig.9	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.



Application No: GB 0006487.3
Claims searched: 17-18

Examiner: Mike Davis
Date of search: 23 March 2001

Patents Act 1977

Further Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK Cl (Ed.S): G4A (AFGDC, AFGDX), G4H (HRCE, HNEC, HNEM, HNP)
Int Cl (Ed.7): G06F, G08C
Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	GB 2309567 A (WATSON) eg abstract and Fig.1	17-18
X	GB 2304952 A (SAMSUNG ELECTRONICS) eg abstract and page 5 line 27 to page 6 line 5	17-18

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.